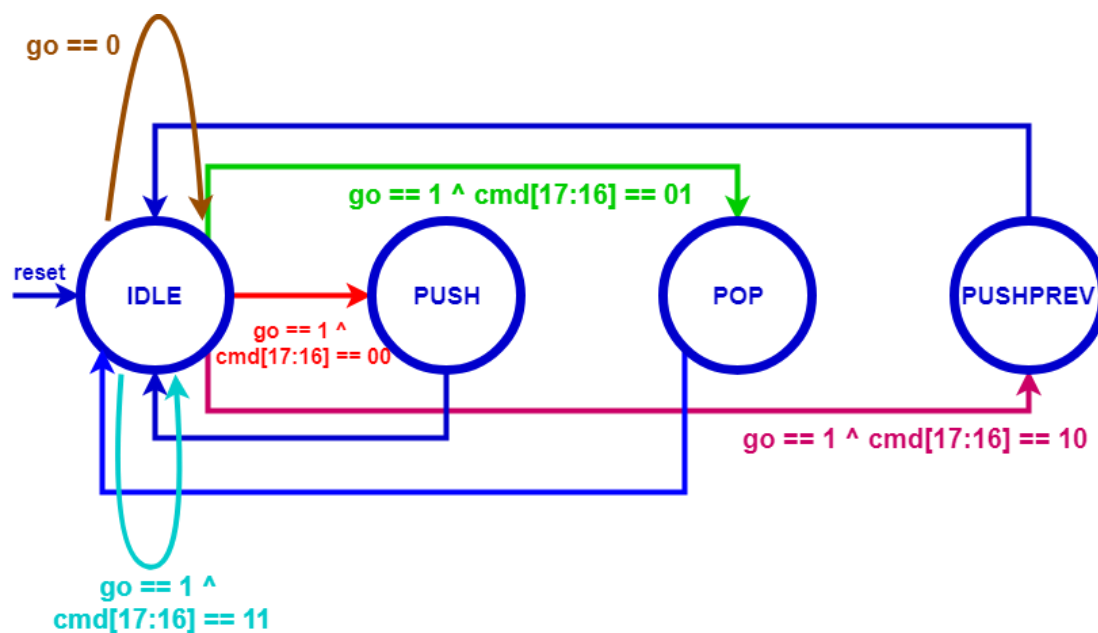


一、FSM



圖(一) block diagram

	IDLE	PUSH	POP	PUSHPREV
ready	1	0	0	0
w_en	0	1	0	1
r_en	0	0	1	0

combinational circuit 共有四個 state，分別是 IDLE、PUSH、POP、PUSHPREV，reset 後進入 IDLE，go == 0 時一直待在 IDLE，接著 go==1 時，依據 cmd 前兩個 bits 決定要進入哪個 state，進入 PUSH、POP、PUSHPREV 運算完後，再回到 IDLE 準備接收下一個資訊。

二、設計過程

主要為兩個 always block，一個為 flip-flop，另一個是 combinational circuit，combinational circuit 有四個 state，分別是 IDLE、PUSH、POP、PUSHPREV。reset==1 時，reset 所有值進入 IDLE，直到 go==1 時，再進入其他 cmd 要求的 state，而每次到 clk 的 posedge，就把 combinational circuit 運算完的所有 next_變數，傳給主要變數。而為了避免還沒運算完成，就被 testbench 檢查變數，ready 的值就擔任很重要的角色，在 IDLE state 時 ready 設為 1，其他三個 state ready 為 0，而每次進到 PUSH、POP、PUSHPREV 後，next_state 設為 IDLE，這樣代表運算時 ready 會是 0，運算完後 ready 又變回 1，這樣就可以在運算後變數才被檢查，而且運算完，可以隨時準備再讀取資料。

三、問題及解決方法

最一開始我打完的時候發現語法錯誤，而且錯誤很奇怪，一直抓不出來，後來花了一個小時才發現是在 define 的地方不需要打分號，可見對語法熟悉是麼重要的事。

一開始 combinational circuit 部分我分成太多個 always block 寫，不方便管理，一直出錯，ready 的地方一直保持是 1，後來我把它合併為一個 always block，整個思緒就變清晰許多，比較好 debug。

剛開始我的 make sim0、1、2 都過了，但 make syn0、1、2 都沒過，原因是 addr 出錯，剛開始的 PUSH 應該要 PUSH 到 ram 的第 0 格，但我都 PUSH 到第 1 格，後來發現是因為我在 combinational circuit 直接把位置的值給 addr，而沒有多設一個 next_addr 變數，在 combinational circuit 部分完成 next_addr 運算，於 positive edge 時再傳值，所以 addr 在一瞬間就被加 1，數值變成 1 了，w_num 則一開始就被存到 ram[1]，以後在寫 sequential circuit 時要更小心才是。

四、模擬、合成結果

```
*Verdi3* Loading libsscore_ius141.so
*Verdi3* : Enable Parallel Dumping.
ncsim> source /usr/cad/cadence/INCISIV/cur/tools/inca/files/ncsimrc
ncsim> run
FSDB Dumper for IUS, Release Verdi3_L-2016.06-SP1-1, Linux, 09/27/2016
(C) 1996 - 2016 by Synopsys, Inc.
*Verdi3* FSDB WARNING: The FSDB file already exists. Overwriting the FSDB file may crash the programs that are using this file.
*Verdi3* : Create FSDB file 'queue.fsdb'
*Verdi3* : Begin traversing the scopes, layer (0).
*Verdi3* : End of traversing.
-----
command: 000000000000000001
-----
command: 000000000000000011
-----
command: 0000000000000000111
-----
command: 010000000000000000
-----
command: 010000000000000000
-----
command: 000000000000001111
-----
command: 000000000000011111
-----
command: 000000000000111111
-----
command: 100000000000000000
-----
command: 010000000000000000
-----
Congratulations!!
Simulation complete via $finish(1) at time 3150 NS + 0
./queue_tb.v:160 $finish;
ncsim> exit
```

圖(二)make sim0

```

command: 010000000000000000
-----
command: 100000000000000000
-----
command: 110000000000000011
-----
command: 000000000000000111
-----
command: 000101001001001010
-----
command: 001010110000010100
-----
command: 000010101111111110
-----
command: 001111111111111111
-----
command: 000000000000000000
-----
command: 001111111111111111
-----
command: 010101001010010101
-----
command: 010000010100100101
-----
command: 010000000010101000
-----
command: 100000000101011000
-----
command: 001010101010000000
-----
command: 110100000000000000

Congratulations!!

Simulation complete via $finish(1) at time 5850 NS + 0
./queue_tb.v:160
ncsim> exit                                $finish;

```

圖(三) make sim1

```

command: 110000000001010100
-----
command: 001111111111111110
-----
command: 001111111111111100
-----
command: 00111111111111000
-----
command: 00111111111111111
-----
command: 00111111111111110
-----
command: 00111111111111100
-----
command: 00111111111111000
-----
command: 00111111111111111
error, this addr isn't empty!!
-----
command: 00111111111111110
error, this addr isn't empty!!
-----
command: 00111111111111100
error, this addr isn't empty!!
-----
command: 00111111111111000
-----
command: 110000000001010100
-----
command: 110000000001010100

Congratulations!!

Simulation complete via $finish(1) at time 33400 NS + 0
./queue_tb.v:160
ncsim> exit                                $finish;

```

圖(四) make sim2

```

Loading snapshot worklib.queue_tb.v ..... Done
*Verdi3* Loading libsscore_ius141.so
*Verdi3* : Enable Parallel Dumping.
ncsim> source /usr/cad/cadence/INCISIV/cur/tools/inca/files/ncsimrc
ncsim> run
FSDB Dumper for IUS, Release Verdi3_L-2016.06-SP1-1, Linux, 09/27/2016
(C) 1996 - 2016 by Synopsys, Inc.
*Verdi3* FSDB WARNING: The FSDB file already exists. Overwriting the FSDB
*Verdi3* : Create FSDB file 'queue_syn.fsdb'
*Verdi3* : Begin traversing the scopes, layer (0).
*Verdi3* : End of traversing.
-----
command: 000000000000000001
-----
command: 000000000000000011
-----
command: 0000000000000000111
-----
command: 010000000000000000
-----
command: 010000000000000000
-----
command: 000000000000001111
-----
command: 000000000000011111
-----
command: 000000000000111111
-----
command: 100000000000000000
-----
command: 010000000000000000

Congratulations!!

Simulation complete via $finish(1) at time 3150 NS + 0
./queue_tb.v:160
ncsim> exit $finish;

```

圖(五) make syn0

```

command: 010000000000000000
-----
command: 100000000000000000
-----
command: 110000000000000011
-----
command: 000000000000000111
-----
command: 000101001001001010
-----
command: 001010110000010100
-----
command: 000010101111111110
-----
command: 001111111111111111
-----
command: 000000000000000000
-----
command: 001111111111111111
-----
command: 010101001010010101
-----
command: 010000010100100101
-----
command: 010000000010101000
-----
command: 100000000101011000
-----
command: 001010101010000000
-----
command: 110100000000000000

Congratulations!!

Simulation complete via $finish(1) at time 5850 NS + 0
./queue_tb.v:160
ncsim> exit $finish;

```

圖(六) make syn1

```

-----
command: 110000000001010100
-----
command: 00111111111111110
-----
command: 001111111111111100
-----
command: 00111111111111000
-----
command: 0011111111111111
-----
command: 0011111111111110
-----
command: 0011111111111100
-----
command: 0011111111111000
-----
command: 001111111111111
error, this addr isn't empty!!
-----
command: 001111111111110
error, this addr isn't empty!!
-----
command: 001111111111100
error, this addr isn't empty!!
-----
command: 001111111111000
-----
command: 110000000001010100
-----
command: 110000000001010100
-----
Congratulations!!

Simulation complete via $finish(1) at time 33400 NS + 0
./queue_tb.v:160                                     $finish;
ncsim> exit

```

圖(七)make syn2